Anomaly Detection in Cellular Networks based on Network Function KPI

Hung-Chin Jang

國立政治大學資訊科學系

jang@cs.nccu.edu.tw

Yu-Wei Chang

國立政治大學資訊科學系

107703034@nccu.edu.tw

Chia-Yu Cheng

國立政治大學資訊科學系

107703001@nccu.edu.tw

**一、摘要**

隨著互聯網的高速發展，網路服務已經快速深入到社會當中，人們大量上網從事購物、付款、娛樂等等活動，因此保障網路的穩定性非常重要。網路服務的穩定性主要靠維運人員監控各式各樣的關鍵性能指標(Key Performance Indicators，簡稱KPI)來判斷網路是否穩定。由於KPI發生異常，往往代表著與其相對的應用或服務發生了問題，若無法在異常發生前將其識別且進行修復，會導致要求呼叫失敗、網路延遲等故障，對全網的服務質量帶來重大的負面影響， 有時甚至等到用戶投訴時才發現異常情況，將會使用戶體驗大大受損。

**二、簡介**

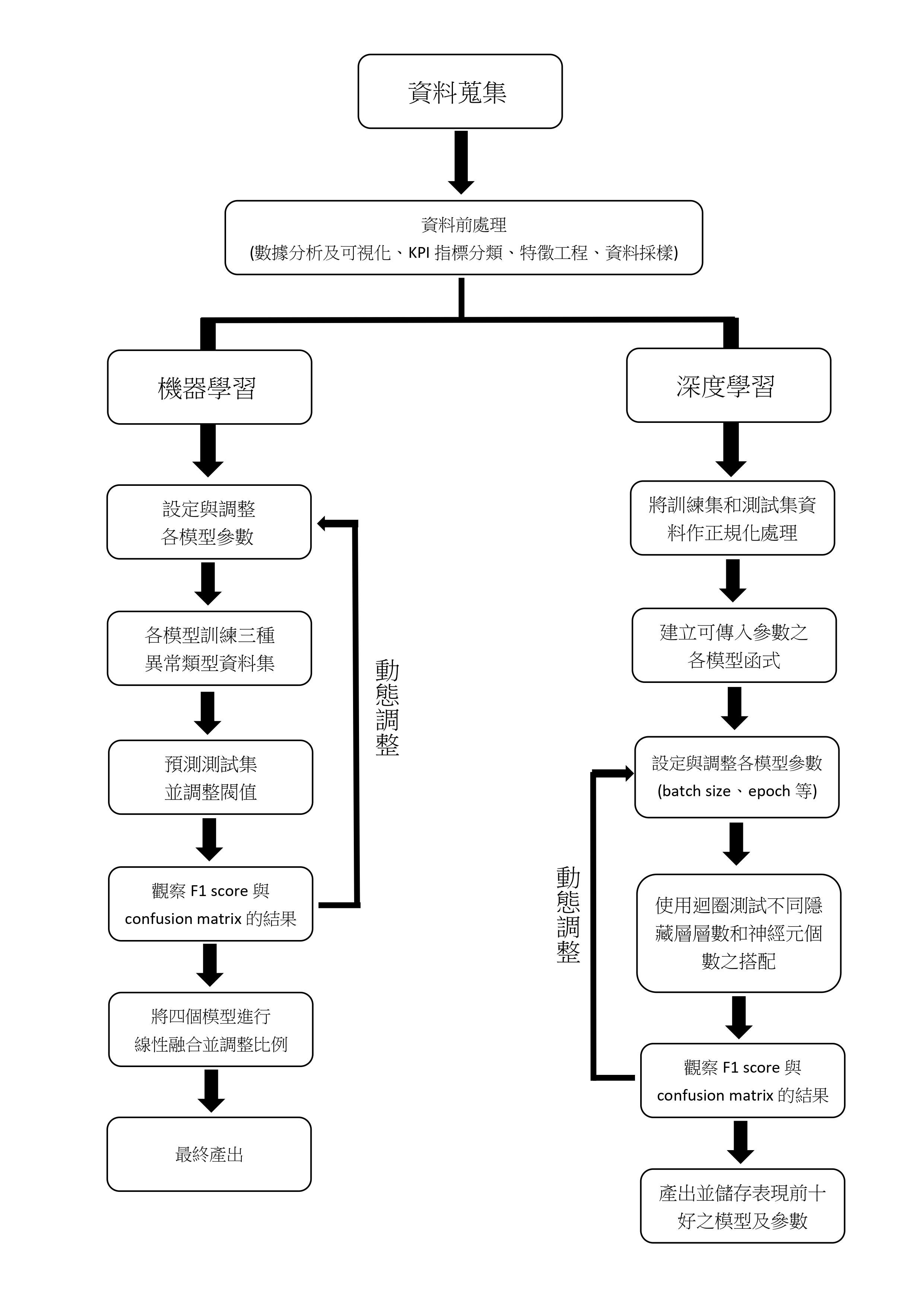
KPI異常檢測指的是透過演算法分析KPI的時間序列數據，並標記網路在何時發生異常，然後利用其時間序列的規律性進行異常預測，如此一來能夠幫助維運人員提前發現未知風險，降低網路因異常帶來的損失。KPI異常預測有以下難點存在：(1) 在實際網路服務場景中，異常發生的頻率相對很低，因此可供分析的異常樣本非常少。(2) 由於核心網網元數據多，並且會不斷更新升級，從而導致異常種類的多樣性。(3) KPI曲線的多樣性，如週期型或波動型等等。因為這些難點，導致大量異常誤報和漏報的存在，如何設計一個高效且具有較強泛化能力的異常檢測模型即是一大挑戰。本研究的實驗數據來自2021華為某次學習賽提供之某網路營運商之部分網元的KPI真實數據，數據採樣間隔為一小時，我們根據歷史一個月的數據訓練各種模型並預測隨後一周內KPI中的異常情況。本文從多維度特徵提取和基於機器學習及深度學習模型建構等方面對單維度的KPI數據進行研究，目的是找到有效的特徵提取方式和異常預測方法，從而增強維運人員對網路服務故障的應對能力。

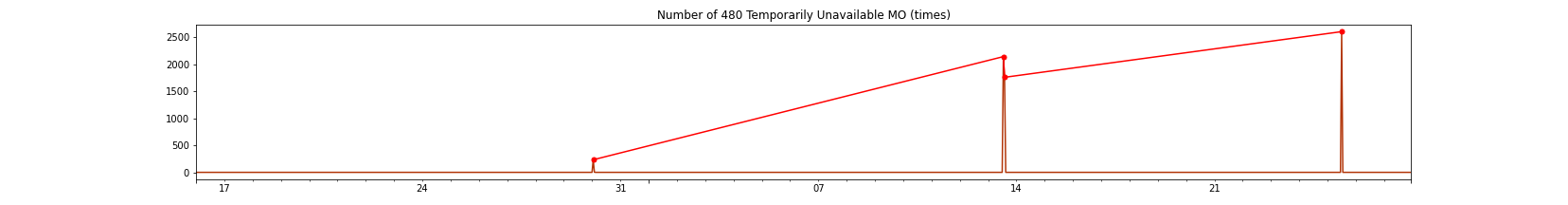
圖1：本研究架構圖

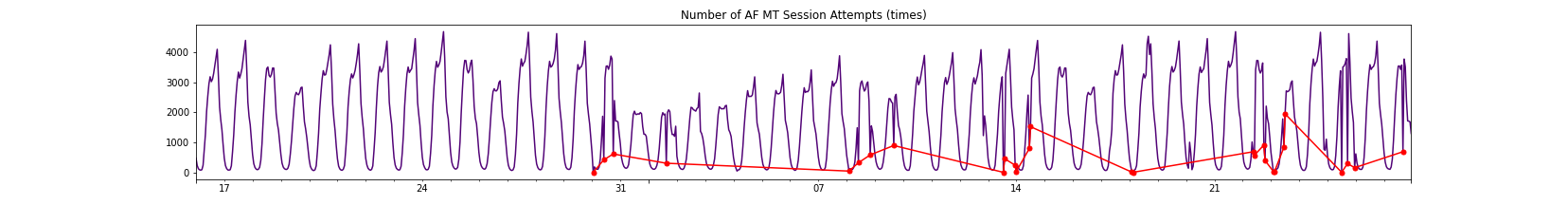
**三、資料前處理**

(一) 數據分析+數據可視化

KPI序列圖對於快速理解資料有著重要的幫助，首先我們將本研究中102個KPI指標分別作圖進行可視化並且加以觀察，最後大致劃分成為以下三種:

(1). 邊界型異常: 邊界型異常如圖2所示，其中異常樣本的取值範圍與正常樣本的取值範圍存在明顯的差異，也就是說存在明確的決策邊界可以劃分出這些異常點。(2). 週期型異常: 週期型異常如圖3所示，整個KPI指標的正常樣本呈現週期性走向，其中異常值的出現往往會偏離這個趨勢走向，但異常值的取值範圍仍然會在正常樣本的取值範圍中，這類異常與其它天同時刻正常點取值的差異會來得較大。(3). 無異常型: 無異常型如圖4所示，此類KPI指標不存在任何異常樣本，對於整體預測的貢獻度不高，於是我們將這類KPI皆預測為正常樣本。

圖2：邊界型異常示意圖

圖3：週期型異常示意圖

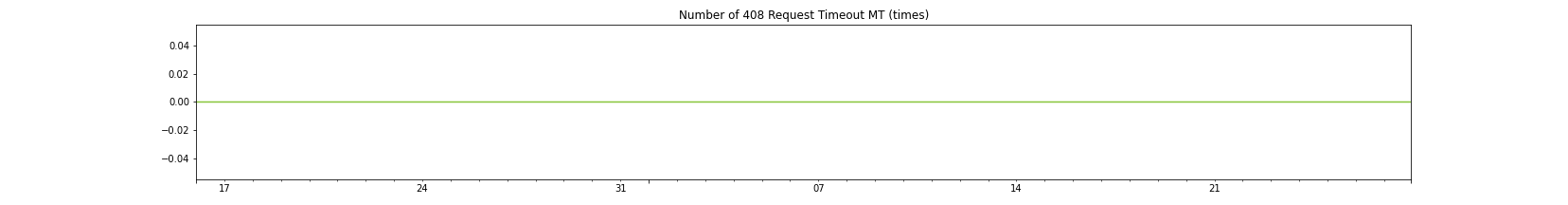


圖4：無異常型異常示意圖

(二) KPI指標分類

由於本數據集的KPI指標共有102個，倘若我們將所有KPI一次拿來建立一個二元分類模型，模型的表現將會差強人意。另一個想法是將每個KPI分別單獨建立模型，但這樣則需要設定並且調參優化至少102個模型，維護成本過於辛苦，因此我們最終決定將KPI指標先進型分類再作模型訓練。

步驟1. 決定邊界型異常: 由於邊界型異常中異常樣本的取值範圍往往偏離正常範圍很大，所以適合用Decision tree這種會根據目標的分布將樣本劃分在不同特徵空間範圍內的方法，來做出邊界的劃分，並挑選出邊界型異常KPI。我們首先將102個KPI依序用Decision tree遍歷，假如只依靠KPI中原生的單一變量(各時間下的KPI數值)建立的簡單Decision tree在訓練集中的F1-score為1，則將它分類為邊界型異常KPI，並且使用該Decision tree模型對相同的KPI的未來樣本進行後續預測。在經過遍歷後，我們獲得30個邊界型異常KPI，然而最終實驗結果也顯示，此做法不僅在訓練集中能100%成功預測邊界型異常，在測試集中也能接近100%預測成功。

步驟2. 決定無異常型: 此步驟相對直觀，我們將KPI中所有Label皆為0(正常樣本)的通通歸納為此類型，並且預測其未來的樣本皆為正常樣本。本次實驗數據中我們總共有29個無異常型KPI。

步驟3. 決定週期型: 沒有分類在前述兩種類型的剩餘43種KPI，我們皆把它歸類為週期型KPI，而我們再對這些週期型KPI進行細分，分組的基本想法便是，週期形狀(趨勢)相似的KPI應該被分在相同組別中，而我們採用Pearson相關係數來分析，其意義是表示兩個變數同向或反向的變動程度，於是非常適合運用在KPI這類時間序列的數據上。通過Pearson相關係數，我們將兩兩相關係數在0.9以上的KPI再細分為同一組。

(三) 特徵工程

由於原始數據中KPI指標的特徵只有一個，即各時間點上的KPI值，若只使用此單一特徵來進行模型訓練，得到的結果將不符合期待，於是我們需要對其生產出更多特徵，使往後步驟中我們有足夠的上限可以使模型和演算法去逼近。本次實驗中我們構造了以下5種類型的特徵: 1. 基礎特徵: 時間、日期、一天中的第幾小時，是否是周末，各個KPI的Label encoder等等。 2. 統計特徵: 計算過去某一段時間範圍內的平均值、中位數、最大值、最小值、方差，分位數和偏度等等。 3. 平移特徵: 前特定某一個時間點KPI的值或差分取值。 4. 差分特徵: 計算當前時間點和上一個時間點或上一個週期中相同時間點的差，差值的大小可以反應出波動大小，觀察是否發生異常波動。 5. 趨勢特徵: 用當前時間點的值減去前1、2、3…等等時間單位時刻的平均值，差值為正，可以反應出上升的趨勢，相反則表示下降的趨勢。本次實驗提取了44種不同特徵供模型訓練使用。

(四) 資料採樣

由於在異常檢測中，異常發生的頻率較低，異常樣本的數據往往很少，我們需要面對的一個挑戰是樣本極度不平衡，如同本次實驗的數據集，異常樣本的比例佔所有資料僅僅0.9%，如果直接將樣本進行訓練，則模型會傾向於將所有樣本都預測為正常樣本，如此便無法達到異常檢測的目的。我們採用SMOTENC，一種資料過採樣演算法，來對數據中樣本較少的那類(本實驗中的異常樣本)進行重複採樣，通過增加分類中少數類樣本的數量來實現樣本均衡。SMOTENC為SMOTE的一種變體，因為SMOTE在採樣時用到了距離，因此當被挑選的數據為含有離散變量的數據(如0為正常，1為異常)，此時無法直接對離散變量數據計算歐式距離，而SMOTENC在處理這類離散變量數據時，會採用K近鄰樣本中出現頻率最高的離散數據作為新樣本的值。最後我們以此種方法對資料進行過採樣，以達到正常異常樣本1:1。

1. **機器學習**

本次實驗我們選擇了四個機器學習模型，其中三種基於GBDT(Gradient Boosting Decision Tree)演算法，分別為XGBOOST、LIGHTGBM和CATBOOST，雖然它們從結構上來說都是決策樹(Decision Tree)，但是樹的特徵及生成過程仍然有很大的不同。其中XGBOOST的決策樹是level-wise增長(同時分裂同一層的葉子)，優點是容易進行平行優化，過擬和(overfitting)的風險較小，缺點則是實際上許多葉子分裂的效益較低，帶來許多較沒必要的負擔。LIGHTGBM的決策樹是leaf-wise增長(每次從所有葉子中尋找分裂效益最大的一個葉子)，缺點是容易長出較深的決策樹，發生overfitting，我們可以透過限制樹的最大深度來解決此一問題。CATBOOST的決策樹採complete binary tree，引入了自動將類別特徵處理為數值型特徵的創新技巧，且透過組合類別特徵，可以利用到特徵之間的關聯，這極大豐富了特徵的維度。最後一種為NGBOOST，使用所謂Natural Gradient Boosting，相較前三者來說，其特別之處是可以預測出機率分佈，這個方法的優點在於，預測的結果並非一個點的估計值，而是給出整個機率分佈，讓每個資料點預測的不確定性可以被考慮進來，有效改善overfitting。

本階段機器學習的步驟如下：

(1). 設定與調整各模型參數

(2). 各模型訓練三種異常類型資料集

(3). 預測測試集並調整閥值

(4). 觀察F1 score與confusion matrix的結果。不斷重複以上四個步驟以獲得三種模型的最終結果。

(5). 將四個模型進行線性融合並調整融合比例以獲得整體最終結果。

本次實驗我們最終以25:10:25:40的比例融合、XGBOOST、LIGHTGBM、

CATBOOST及NGBOOST，並將閥值設定在0.4獲得將近0.8的F1 score。

從實驗結果上可以觀察到，在邊界型異常及無異常樣本上預測的正確率相當高(約95%)，而大部分未能準確預測的樣本大多為週期型異常，由此可知在KPI異常偵測中，要在呈現規律周期的KPI指標內預測出異常是最難突破的一個問題，或許除了KPI值此單一特徵外，還需要更多額外資訊輔助才能有效的在週期型異常上獲得更佳的預測表現。

**實驗結果**

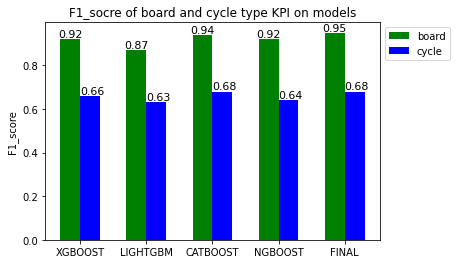


圖5：各模型在邊界型異常和週期型異常上之表現

圖5顯示每個模型分別在邊界型異常和週期型異常上的預測結果之F1-score，可以發現所有模型在週期型異常的表現上皆不如邊界型異常，推測是由於週期型異常的資料點仍在正常值的取值範圍內，所以這些異常點難以和其他正常點明顯的被區分。但就邊界型異常而言，已經可以達到很優秀的預測結果。

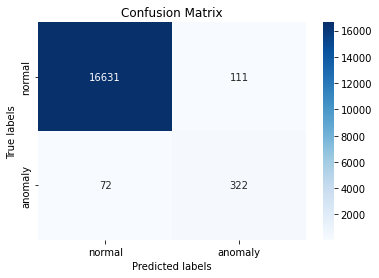


圖6：融合後模型的混淆矩陣

圖6顯示最終線性融合後模型的混淆矩陣，其中FN(左下角)為「被預測為正常樣本的異常樣本」，而這些被預測錯誤的樣本中有10筆來自邊界型異常，62筆來自週期型異常，由此也可知在預測週期型異常上仍頗具難度。FP(右上角)則為「被預測為異常樣本的正常樣本」，在調整模型融合比例的時候，我們也嘗試出其他可以獲得相似F1-score的組合，這些組合通常FN>72，而FP<111，而在實際場景中，比較著重於較低的FN，因為我們希望只要有異常狀況都可以被預測出來並盡早採取預防措施，才不會使其危害整個網路運作，影響用戶體驗。而FP相對較高雖然會導致誤報數量較多，但考量其較不會危害到整體網路品質，於是我們最終選擇FN相對低的此種融合比例作為最終結果。

**五、深度學習**

本部分使用三種神經網路模型進行建模：

(1). 深層神經網路DNN,(Deep Neural Network)：

又稱為全連接神經網路，全連接神經網絡通常有多個隱藏層，每個結點和下一層所有結點都有運算關係，因此它的參數最多，計算量最大。增加隱藏層可以更好分離數據的特徵，但過多的隱藏層也會增加訓練時間以及產生過擬合。

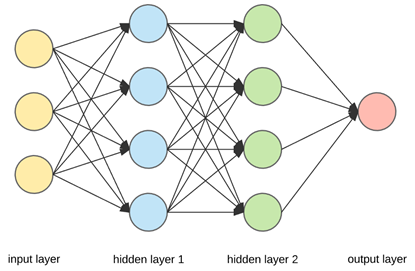


圖7：DNN構造示意圖

(2). 循環神經網路RNN（Recurrent neural network）之衍生模型：

(a)長短期記憶LSTM(Long Short-Term Memory)：

長短期記憶（Long short-term memory, LSTM）是一種RNN的衍生模型。RNN的下個輸出結果與前一個時間點的資料有關聯，但由於普通的RNN無法處理已經為時過久的訊息，導致越久之前的訊息對當下的影響力越下，進而導致訓練過程中的梯度消失和梯度爆炸問題，而LSTM模型能改善這一缺點。

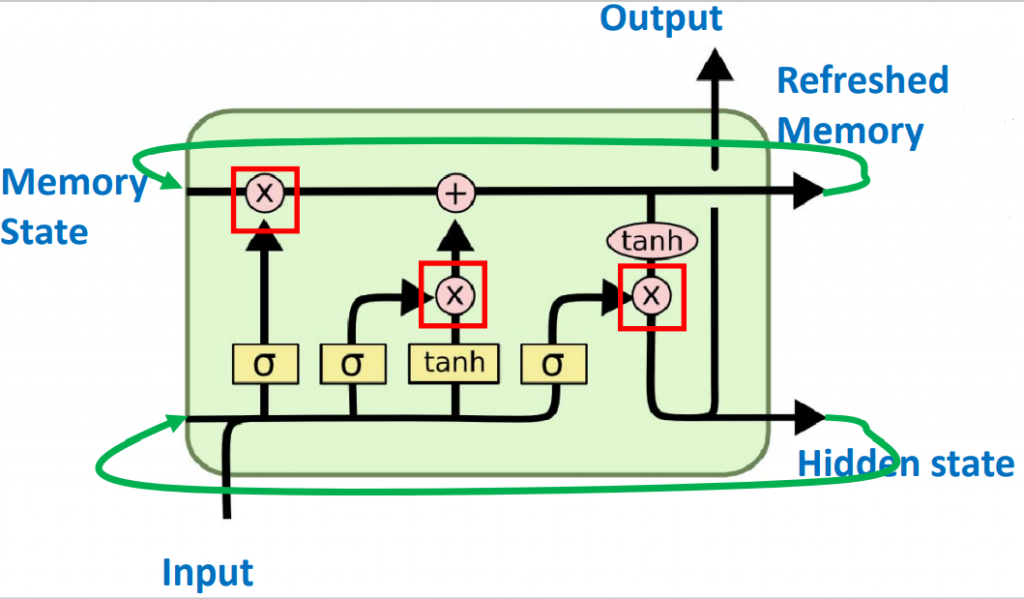


圖8：LSTM構造示意圖

相較於一般的RNN，LSTM的memory更為複雜，存在著3個gate，4個input，1個output。

※ 3個Gate：

1. Input Gate: 某個neuron的output想寫入Memory Cell之前，要先經過input

gate，超過閥值才有辦法把值寫入Memory Cell。

2. Output Gate: 決定是否允許其它neuron讀取Memory Cell的值。

3. Forget Gate: 決定什麼時候Memory Cell清空。

※ 4個Input：

1. 想寫入Memory Cell的input

2. 操控Input Gate的訊號

3. 操控Output Gate的訊號

4. 操控Forget Gate的訊號

(b) GRU (Gate Recurrent Unit)：

GRU (Gate Recurrent Unit)也是一種RNN的衍生模型，內部構造與LSTM相似，但與LSTM相比，GRU把LSTM中的forget gate和input gate用update gate來替代，GRU內部少了一個gate，參數也比LSTM少。

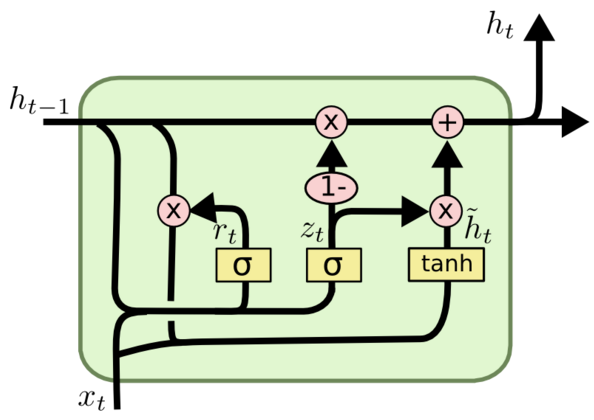


圖9：GRU構造示意圖

由於本研究旨在預測數據異常點，使用Accuracy當作模型優劣判斷指標較為不妥，因為數據異常點個數相較正常點個數少了許多，當模型Accuracy很高時，背後的意涵可能代表的是模型將大部分的正常點和本研究所欲找出的異常點都判斷為正常點，與研究的目的不符。因此，我們改用F1-score指標來判斷模型的好壞。F1-score牽涉到混淆矩陣(Confusion Matrix)、精確率（Precision）以及召回率（Recall）。

混淆矩陣(Confusion Matrix)：

1. TP：預測為真(P)，預測正確(T)。

2. TN：預測為假(N)，預測正確(T)。

3. FP：預測為真(P)，預測錯誤(F)。

4. FN：預測為假(N)，預測錯誤(F)。



圖10：混淆矩陣

準確率（Accuracy）= (TP+TN) / (TP+FP+FN+TN)  
精確率（Precision）= TP / (TP+FP)，即真的樣本中有幾個是預測正確的。  
召回率（Recall）= TP / (TP+FN)，即事實為真的樣本中有幾個是預測正確的。  
F1-score = 2 / ( (1/ Precision) + (1/ Recall) )，即精確率與召回率的調和平均數。

本次實驗我們選擇了三個深度學習模型，分別為DNN、LSTM和GRU，這三種都是神經網路模型，都包含了輸入層、輸出層以及多層隱藏層。在三個模型中，都選擇ReLU來作為激活函數，主要原因是相較於 sigmoid、tanh等激活函數，ReLU較能夠解決梯度消失問題。而所謂的梯度消失問題，指的是當建立一個深層的類神經網路時，每個類神經元會隨訓練過程更新。在接近輸出層網路的類神經元，更新速度會趨近正常，但在接近輸入層的類神經元，其更新速度緩慢，可能內部參數與初始化的參數相同。由於接近輸入層的網路無法作用，讓這個深度學習網路僅有少數的層在運作，便失去深度學習的意義。

優化器的部分選擇Adam，因為Adam相較於其他優化器，更適合大數據集和高維空間的資料，這與我們的資料相符合。而在損失函數的部分，由於此次實驗我們是要分辨異常點出現的時機，資料集value的部分0代表的是正常點，1代表的是異常點，屬於二元分類的問題，因此我們選用適合二元分類問題的binary cross entropy作為模型的損失函數。

建立神經網路模型具體實施之四步驟:

步驟一:將訓練集資料和測試集資料作正規化處理

步驟二:建立可傳入參數之各模型函式

步驟三:設定與調整各模型參數(batch size、epoch等)

步驟四:使用迴圈測試不同隱藏層層數和神經元個數之搭配

步驟四:針對實驗結果，調整不同的參數。

**實驗結果**

實驗過程中，我們發現在模型的表現上LSTM和GRU遠遠不如DNN，前兩個模型訓練出來最好的F1-score值大約都落在0.2左右，而DNN則能穩定地維持在0.2以上，最高甚至能接近0.5。因此，實驗後期我們選擇了DNN模型作為我們的訓練對象。針對上述兩種RNN類型的模型表現低落的情況，我們推估原因可能有二。一是資料的不平衡，在做資料前處理的資料採集時，雖然表面上看似讓正常和異常的樣本達到1:1，但可能實際上產生的異常樣本不是那麼正確，導致正常和異常的樣本實際上比例並沒有到達平衡，而導致訓練的錯誤。二是樣本的正常或異常或許跟時間序列沒有關係，所以使用擅長預測時間序列問題的RNN模型效果會較不理想。

下圖為1到6層隱藏層數和300到1200神經元個數之參數搭配之F1-score為0.4以上的三維空間顯示圖。從圖中能發現，紅點集中在趨向少層數、多神經元個數的位置。越少的隱藏層層數搭配越多的神經元個數，模型的表現較好。反之，越多的隱藏層層數搭配越少的神經元個數，模型的表現較差。

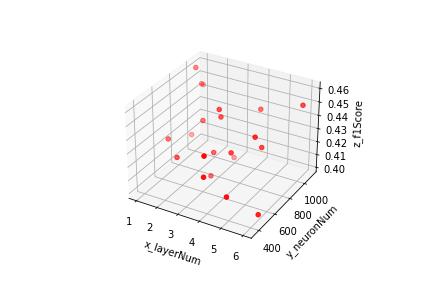


圖11：實驗參數成果F1-score為0.4以上之三維空間顯示圖

**參考文獻**

[1] B. Hussain, Q. Du and P. Ren, "Deep Learning-Based Big Data-Assisted Anomaly Detection in Cellular Networks," 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647366.

[2] Q. Sun and Z. Ge, "Deep Learning for Industrial KPI Prediction: When Ensemble Learning Meets Semi-Supervised Data," in IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 260-269, Jan. 2021, doi: 10.1109/TII.2020.2969709.